

ENGR101: Introduction

Introduction to ENGR101

ECS, VUW

What it is about?

- ① What engineering is about?
 - ① Know what you doing
 - ② Convert this knowledge into working product
- ② Difference with other courses - we focus more on "convert" part
 - Less lecture material. More reading and self-study.
 - More learn-by-doing (and **asking a lot of questions**)
 - Course made out of several projects

Big picture

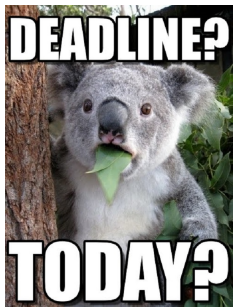
- First two projects - 3 weeks each
 - One week of lectures - no A labs, but there are tutorials during Lab B slot
 - Two weeks of doing the project: one lecture per week, mostly answering the **questions**
 - If in doubt what is happening this week - have a look at Lecture Schedule page: https://ecs.wgtn.ac.nz/Courses/ENGR101_2024T1/LectureSchedule
- Last project - 4 weeks
 - As close to real-life project as we can make it
 - Project is big - requires planning
 - Work in teams - code sharing, version control, Git
 - Combination of software (mostly) and hardware
 - Same mark for all members of the team

Convert knowledge into product - how to start?

- To start the project from the scratch is hard but necessary.
- Code template for Project 1 - just fill in the gaps. Smaller template for Project 2. No code templates for Project 3.
- Minimize distractions:
 - headphones are OK. We will take headphones as a sign that you don't want to be interrupted.
 - disconnect. Use internet only for work.
 - Quiet, please. Other people can be working.
- Don't beat your head against the wall for too long. If you are stuck - ask **questions**. Put in some effort first.
- Start with something simple. Persist. After a while you will get into **zone**. [https://en.wikipedia.org/wiki/Flow_\(psychology\)](https://en.wikipedia.org/wiki/Flow_(psychology))

How to get working code?

- 2 weeks = 14 days = 336 hours. Right, does not look too hard, can make it in three hours!
- Which 3 hours out of 336? Last ones before submission due, of course!
- You will be under stress, produce bad code and programming will be nothing but a grind for you.



End code product

- Only result counts - some of your code will be marked by automated system
- There are pros and contras in it - we don't mark style(pro). If your code does not run (on ECS machines where marking happens) - markers are not going to fix it.
- Our hope is that you will come to conclusion that well-structured code is easier to write. To this end code we will require is bigger (compared with other courses).
- Result should be on time - otherwise customer walks away.

Technical

- Language used - C++ (your granddad favourite language). Known to be dangerous - think ahead before typing the code. WHY????
- OS - we use Linux. Windows and Mac laptops are (should be) fine. Try to keep submitted code standard without OS specific features - it will be marked on Linux machines.
- Basically you can use whatever software you like - we are after working code.

Using automated code generators

You can be aware that there are AI programming tools available - chatGPT, Copilot and likes. Can you use these tools?

Likely that you will have to use these tools when working, so answer is YES.

Is it a magic bullet? NO, far from it. Queries to AI have to be very specific. Code produced by AI often is wrong.

To be productive with these tools you need specific skills. By look of it, typing in the pages of clever code is not viable career path any more. Designing software (of which writing code is only rather small part, like 10-20 percent), on another hand, is unlikely to be automated any time soon.

Questions?